

# Build an HTML Email Template From Scratch



The best way to understand any process is to carry it out yourself, from the ground up. Today, we're going to do just that with email design, by building an HTML email template from scratch.

## Kick Things Off

To begin with, it's worth mentioning where I pulled some of the resources from.

- The lovely 2D icons are by [Pierre Borodin](#) on Dribbble
- The typefaces used are [Oil Can](#), [Source Sans Pro](#) and [Mission Script](#)
- The social media icons are from [Metrize Icons](#)

Now, as we discussed in the [previous tutorial](#), you'll need to begin your HTML email document with an XHTML doctype:

```
1<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
2"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
3<html xmlns="http://www.w3.org/1999/xhtml">  
4<head>  
5<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
6<title>Demystifying Email Design</title>  
7<meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
8</head>  
9</html>
```

With that sorted, we can commence building the rest of the structure.

## Creating the Body and Main Table

First, we'll add an overall structure for our email, starting with a `<body>` tag. We'll set the margin and padding on the body tag to zero to avoid any unexpected space.

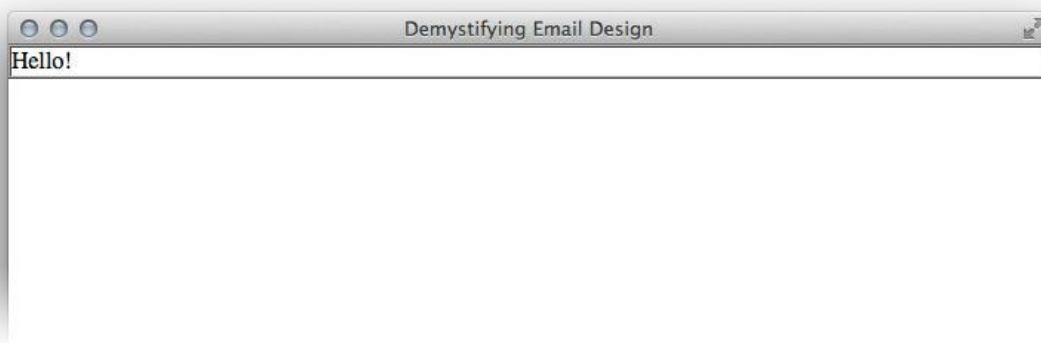
We'll also add a table with a width of 100%. This acts as a true body tag for our email, because styling of the body tag isn't fully supported. If you wanted to add a background color to the 'body' of your email, you'd need to apply it to this big table instead.

Set your cellpadding and cellspacing to zero to avoid any unexpected space in the table.

**Note:** We're going to leave `border="1"` on all of our tables, so that we can see the skeleton of our layout as we go. We'll remove them at the end with a simple Find & Replace.

```
<body style="margin: 0; padding: 0;">

<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td>
Hello!
</td>
</tr>
</table>
</body>
```



If an attribute exists in HTML, use that instead of CSS

Now place a centered table of 600 pixels wide **inside the container table**. 600 pixels is a safe maximum width for your emails to display comfortably within most desktop and webmail clients on most screen resolutions.

Set this width using HTML instead of CSS, by using the width attribute. The golden rule in HTML email development is: if an attribute exists in HTML, use that instead of CSS.

We'll replace our little 'Hello!' greeting with this table:

```
<table align="center" border="1" cellpadding="0" cellspacing="0" width="600" style="border-collapse: collapse;">
<tr>
<td>
Hello!
```

```
</td>
</tr>
</table>
```

We've also added an inline style property that sets the border-collapse property to collapse. If we don't do this, newer versions of Outlook will add a small space between our table and our border.



## Creating the Structure and Header

In our design we can see that the email is divided into three logical sections, so we'll create a row for each.

Let's duplicate the single row we already made so that we have three in total. I've changed the text inside them so that we can easily identify each row.

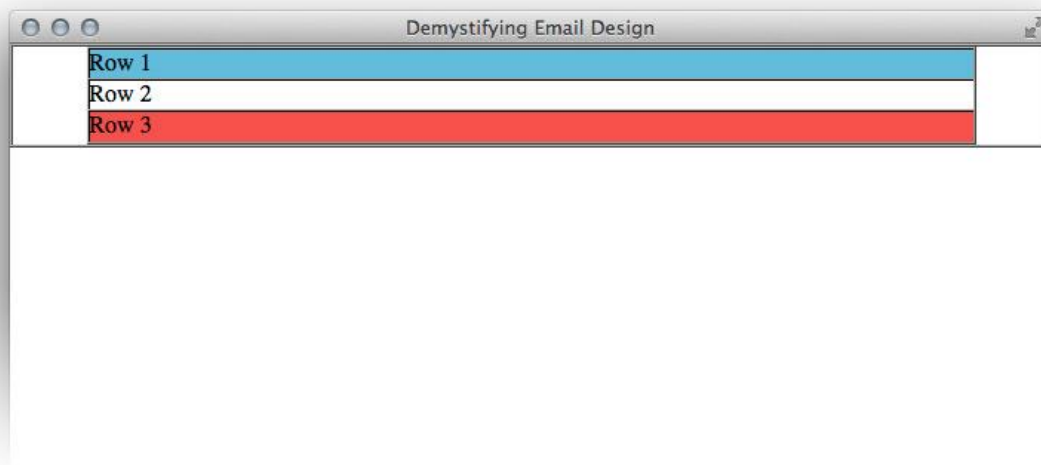
```
<table align="center" border="1" cellpadding="0" cellspacing="0" width="600">
<tr>
<td>
Row 1
</td>
</tr>
<tr>
<td>
Row 2
</td>
</tr>
<tr>
<td>
Row 3
</td>
</tr>
</table>
```



Row 1	
Row 2	
Row 3	

Now we'll color them according to the design. As bgcolor is a valid HTML attribute, we'll use that to set the background color instead of CSS. Always remember to use the full six characters of a hex code, as three character shorthand won't always work.

```
<table align="center" border="1" cellpadding="0" cellspacing="0" width="600">
<tr>
  <td bgcolor="#70bbd9">
    Row 1
  </td>
</tr>
<tr>
  <td bgcolor="#ffffff">
    Row 2
  </td>
</tr>
<tr>
  <td bgcolor="#ee4c50">
    Row 3
  </td>
</tr>
</table>
```



Row 1	
Row 2	
Row 3	

Ok, next up we are going to focus on Row 1. We want to adjust the padding on the cell and then insert our image.

## Using Padding

When using padding in email, you must always specify every single value (top, right, bottom and left) otherwise you can get unpredictable results. I find that you can still use the shorthand, i.e. padding: 10px 10px 8px 5px;, but if you are having trouble you may wish to write it out longform, i.e. padding-top: 10px; padding-right: 10px; padding-bottom: 8px; padding-left: 5px;.

If you are having even greater troubles with padding (for example, if your send platform is stripping out your CSS), don't use it at all. Simply use empty cells to create space. There is no need to use spacer GIFs, just make sure you add style="line-height: 0; font-size: 0;" to the cell, place an &nbsp; inside and give it an explicit height or width. Here is an example:

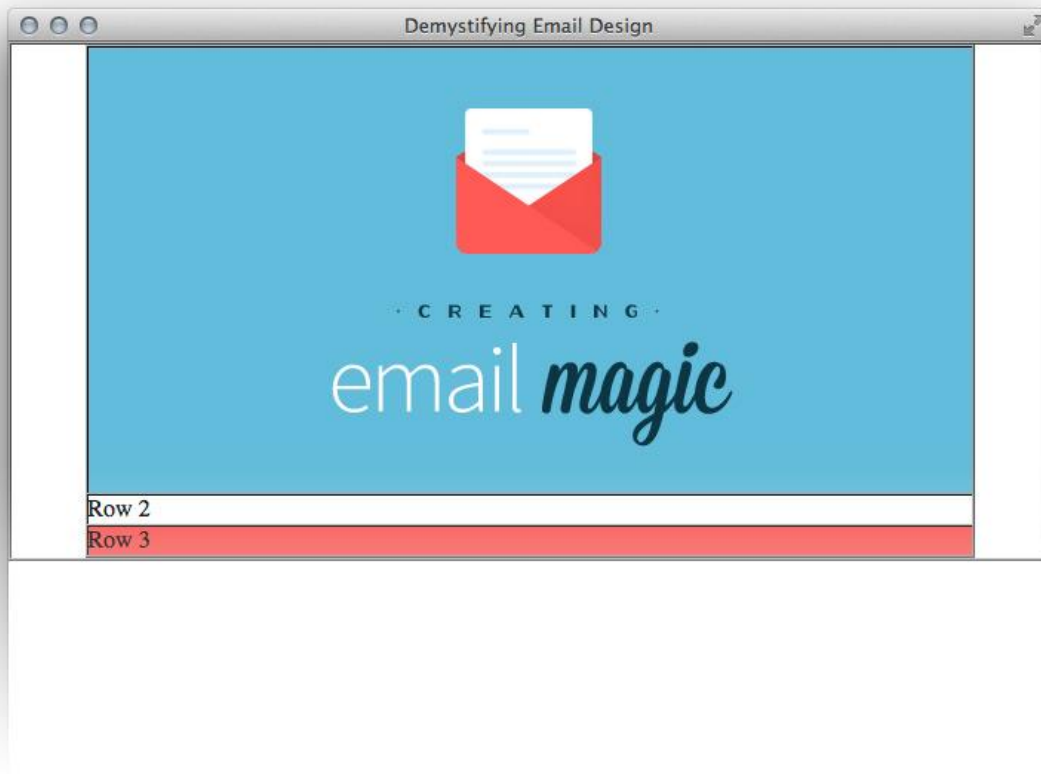
```
1<tr><td style="font-size: 0; line-height: 0;" height="10">&nbsp;</td></tr>
```

Also note that it's safe to use padding on TD tags but not on P tags or DIVs. They behave a lot more unpredictably.

So, we'll use some inline CSS to add padding to the cell. Then we'll insert our image, adding alt text and adding style="display: block;" which is a common fix that stops some email clients adding gaps underneath your images. We'll center the image by adding align="center" to our td tag. We'll also add an alt tag which is important for when our email is initially loaded which, in most cases, will be with images off.

**Note:** If the contents of your header are really important to your message, don't use an image-only header. Remember, images are blocked by default for most clients, so if there is an aspect of your email that is crucial, never include it as an image. In this example, however, my header is pretty superfluous.

```
<td align="center" bgcolor="#70bbd9" style="padding: 40px 0 30px 0;">
  
</td>
```



## Creating the Content Area

First off, we'll add some padding to the middle cell so that the table inside has some space around it, as per our design.

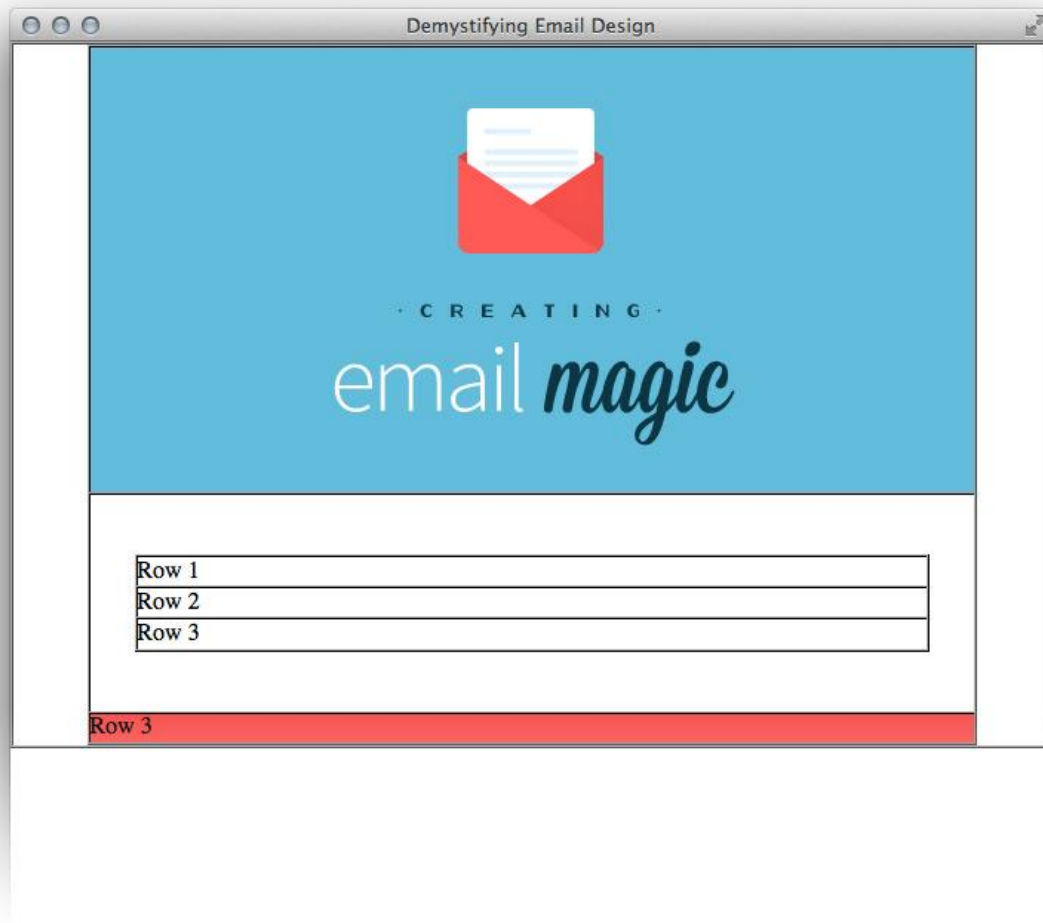


Now we'll add a table with three rows for our main content — one for the headline, one for the introductory text, and one for the row with two columns. We'll set the width of this table to 100% rather than using a pixel value because this will help us further down the track if we want to make our email responsive. If you always have pixel widths on everything, you can end up with a lot of values to override with media queries. If your nested table widths are based on percentages, then when you adjust the width of the parent element, everything will adapt accordingly.

```
<td bgcolor="#ffffff" style="padding: 40px 30px 40px 30px;">
<table border="1" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td>
      Row 1
    </td>
  </tr>
  <tr>
    <td>
      Row 2
    </td>
  </tr>
  <tr>
    <td>
      Row 3
    </td>
  </tr>
</table>
```



```
</td>  
</tr>  
</table>  
</td>
```



Now we'll add in our content, and add some padding to the middle cell.



```
<table border="1" cellpadding="0" cellspacing="0" width="100%">
```

```
<tr>
```

```
<td>
```

```
  Lorem ipsum dolor sit amet!
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="padding: 20px 0 30px 0;">
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```
  Row 3
```

```
</td>
```

```
</tr>
```

```
</table>
```

Now we are going to add our two columns of content to Row 3. Because we want a 'margin' in between these two cells, but margin isn't supported, we'll create a three-column table with an empty cell between the two outer columns.

As much as I like to stick to percentages, when you have content that is a specific size, it can be tricky to convert it to a percentage (in this example, the columns would be 48.1% which could become confusing). For this reason, since our two images are 260px wide, we'll create columns that are also 260px wide, with a 20px margin cell in the middle. (This will total 540px, which is the 600px width of our table minus the padding of 30px on either side.) Be sure to zero your font-size and line-height and add a non-breaking space character `&nbsp;` in the margin cell.

We'll also set the `valign` to "top" for both cells so that they will vertically align to the top, even if one column has more text than the other. The default vertical alignment is "middle".

```
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td width="260" valign="top">
  Column 1
</td>
<td style="font-size: 0; line-height: 0;" width="20">
  &nbsp;
</td>
<td width="260" valign="top">
  Column 2
</td>
</tr>
</table>
```



Now let's add our images and content to those columns. As we need multiple rows, we'll nest yet another table because we can't use any colspan or rowspan tags. We'll also add some padding in between the image and copy in each column.

```

<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td width="260" valign="top">
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td>

</td>
</tr>
<tr>
<td style="padding: 25px 0 0 0;">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit
    ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere
    mauris neque at erat.
</td>
</tr>
</table>
</td>
<td style="font-size: 0; line-height: 0;" width="20">
    &nbsp;
</td>
<td width="260" valign="top">
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td>

</td>
</tr>
<tr>
<td style="padding: 25px 0 0 0;">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit
    ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere
    mauris neque at erat.
</td>
</tr>
</table>
</td>
</tr>
</table>

```

Here we have set the width of the images using HTML at 100% of the column width. This, again, is so that if we make this email responsive, we only have to use media queries to change the width of the parent element. We'll have to override the height in pixels because using `style="height: auto"` now won't work in everything (*cough cough, Outlook*). So we'll set it using pixels. This means we'd have to set `height: auto`!important on those images using media queries to override the pixel value, but we could do that with a single class. As we set the width as a percentage, we won't need to override that. The fewer things to override, the better.



· C R E A T I N G ·

# email *magic*

Lorem ipsum dolor sit amet!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

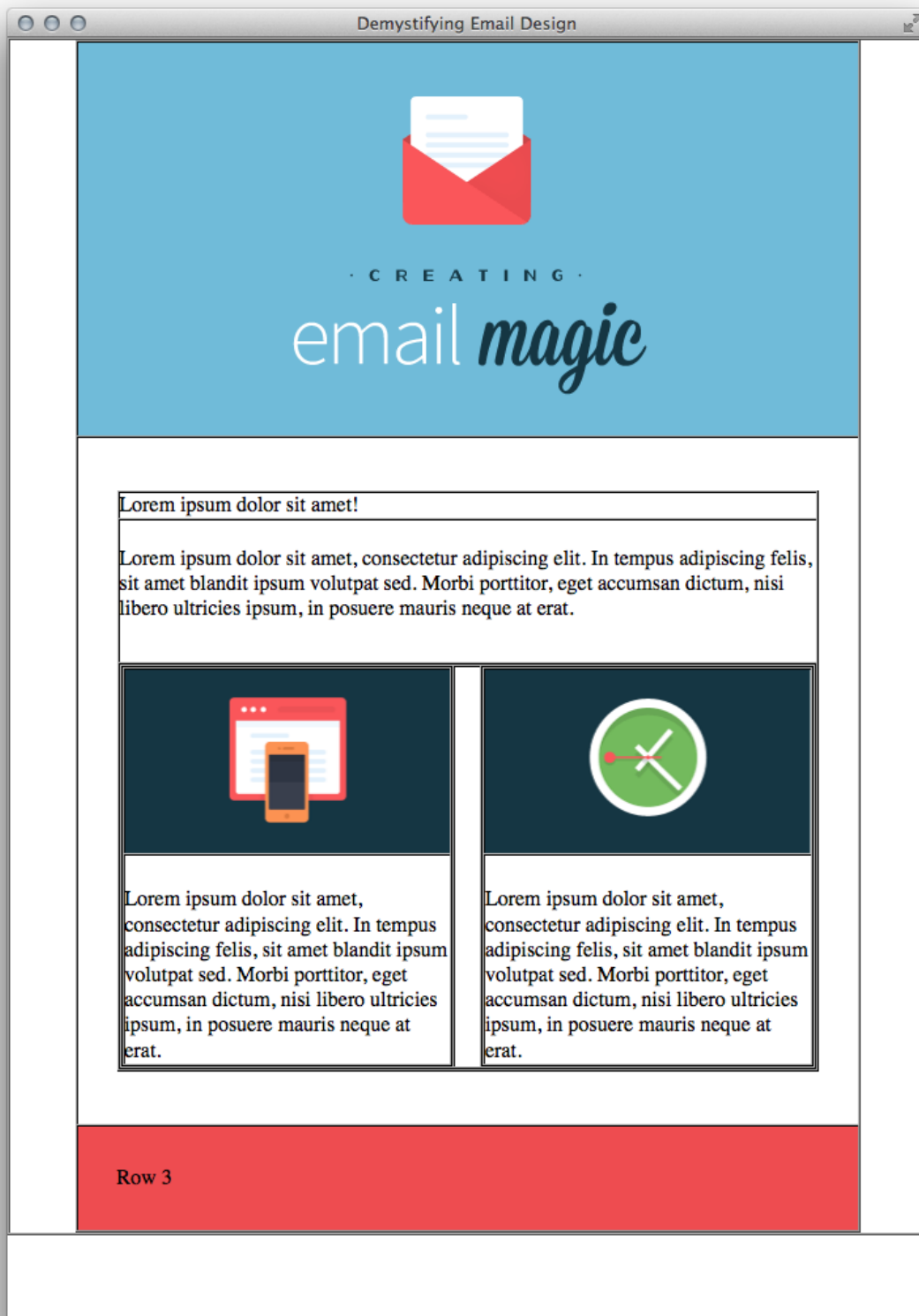


Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

## The Footer

Now we'll add our padding to the footer row.

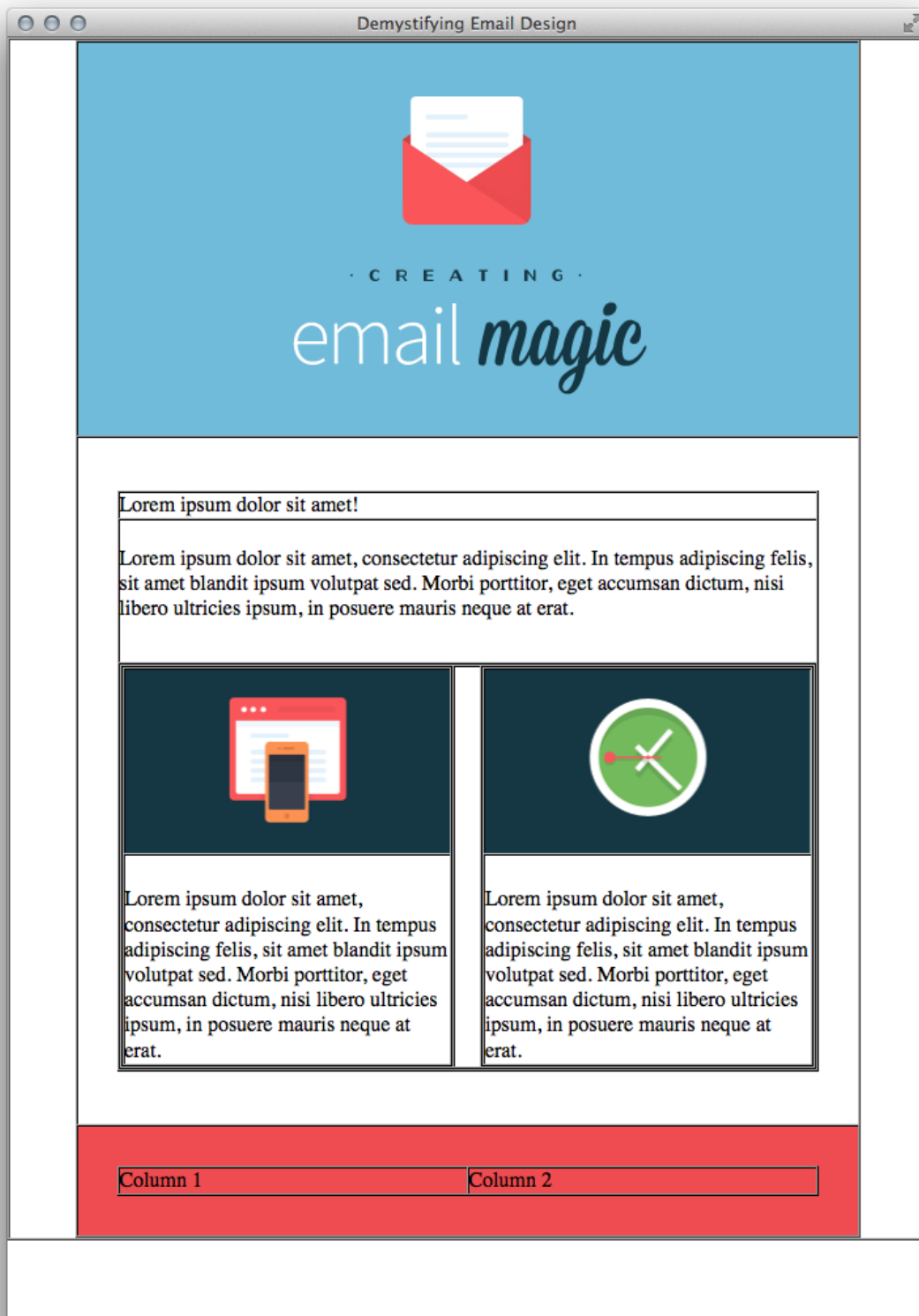
```
<td bgcolor="#ee4c50" style="padding: 30px 30px 30px 30px;">  
Row 3  
</td>
```





Inside that cell, we'll nest another table to get our two columns.

```
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td>
Column 1
</td>
<td>
Column 2
</td>
</tr>
</table>
```



We'll create another little table for our social media icons. We'll set its parent cell to align="right". Make sure you set border="0" on these image links (to avoid a blue link border) and don't forget display:block.

```
<td align="right">
<table border="0" cellpadding="0" cellspacing="0">
<tr>
<td>
<a href="http://www.twitter.com/">

</a>
</td>
<td style="font-size: 0; line-height: 0;" width="20">&nbsp;</td>
<td>
<a href="http://www.twitter.com/">

</a>
</td>
</tr>
</table>
</td>
```



· C R E A T I N G ·  
email *magic*

Lorem ipsum dolor sit amet!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

@ Someone, somewhere 2013  
[Unsubscribe](#) to this newsletter instantly



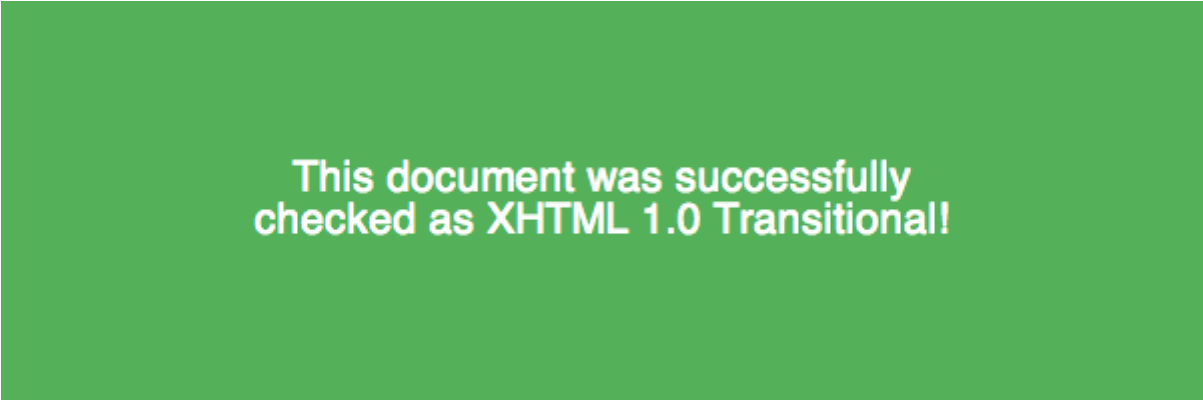
Now we'll add our text and add a width to our cells, just to be safe, even though there is a lot of whitespace between them. We'll set this cell to 75% and the other to 25%.

```
<td width="75%">  
&reg; Someone, somewhere 2013<br/>  
Unsubscribe to this newsletter instantly  
</td>
```

And there we have it! Our layout is complete.

## Validation

Let's run this through the [W3C Validator](#) to make sure nothing is bad or broken. If you've followed along exactly, it will say that it has passed.

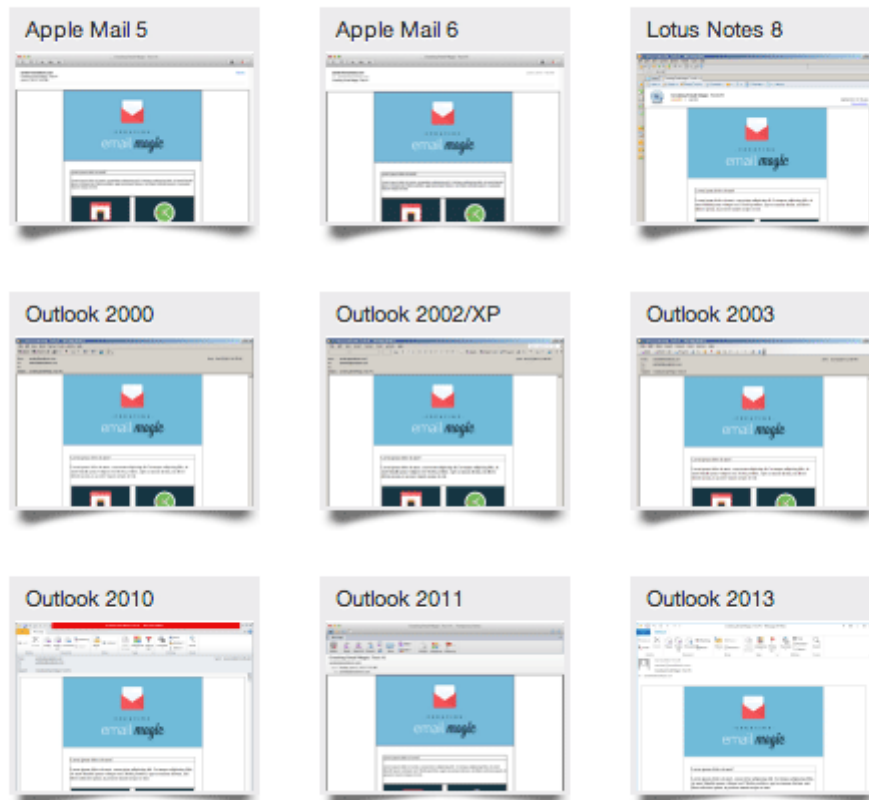


**This document was successfully  
checked as XHTML 1.0 Transitional!**

Next we're going to run a test through Litmus to make sure the structure of our email works great. Here's a summary of my test:

## Desktop Email Clients

---



## Styling our Text

Our first row is the heading. We'll use the `<b>` tag to create bold text because, as we already know, if it exists in HTML, we use that instead of CSS.

```
<td style="color: #153643; font-family: Arial, sans-serif; font-size: 24px;">  
<b>Lorem ipsum dolor sit amet!</b>  
</td>
```

We'll also add this inline style to all the other cells of text:

```
style="color: #153643; font-family: Arial, sans-serif; font-size: 16px; line-height: 20px;"
```

Next we need to style the footer text, and we'll also tidy up our unsubscribe link. We'll style our unsubscribe text link using both CSS and the HTML `<font>` tag. This doubling up is the best way to ensure that your links never show up in the default blue.

```
<td style="color: #ffffff; font-family: Arial, sans-serif; font-size: 14px;">
  &reg; Someone, somewhere 2013<br/>
  <a href="#" style="color: #ffffff;"><font color="#ffffff">Unsubscribe</font></a> to this newsletter
  instantly
</td>
```



· C R E A T I N G ·

# email *magic*

## Lorem ipsum dolor sit amet!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

© Someone, somewhere 2013  
[Unsubscribe](#) to this newsletter instantly





And there we have it! Everything is in. Time to turn off the borders and see it looking beautiful. Go through and replace every occurrence of `border="1"` with `border="0"`.



· C R E A T I N G ·  
email *magic*

## Lorem ipsum dolor sit amet!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

© Someone, somewhere 2013  
[Unsubscribe](#) to this newsletter instantly



At this point, it's looking a little sad floating in white space, so let's go up to our first 600px wide table and add:

```
style="border: 1px solid #cccccc;"
```

Now it doesn't look like it's floating anymore. As a final touch, I'm going to add 30px of padding to the bottom of the very first cell, to prevent our email from stopping abruptly at the bottom in some webmail clients (like Apple Mail), and 10px of padding at the top, so that our blue header has a little bit of breathing room.

```
<td style="padding: 20px 0 30px 0;">
```



· C R E A T I N G ·

# email *magic*

## Lorem ipsum dolor sit amet!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

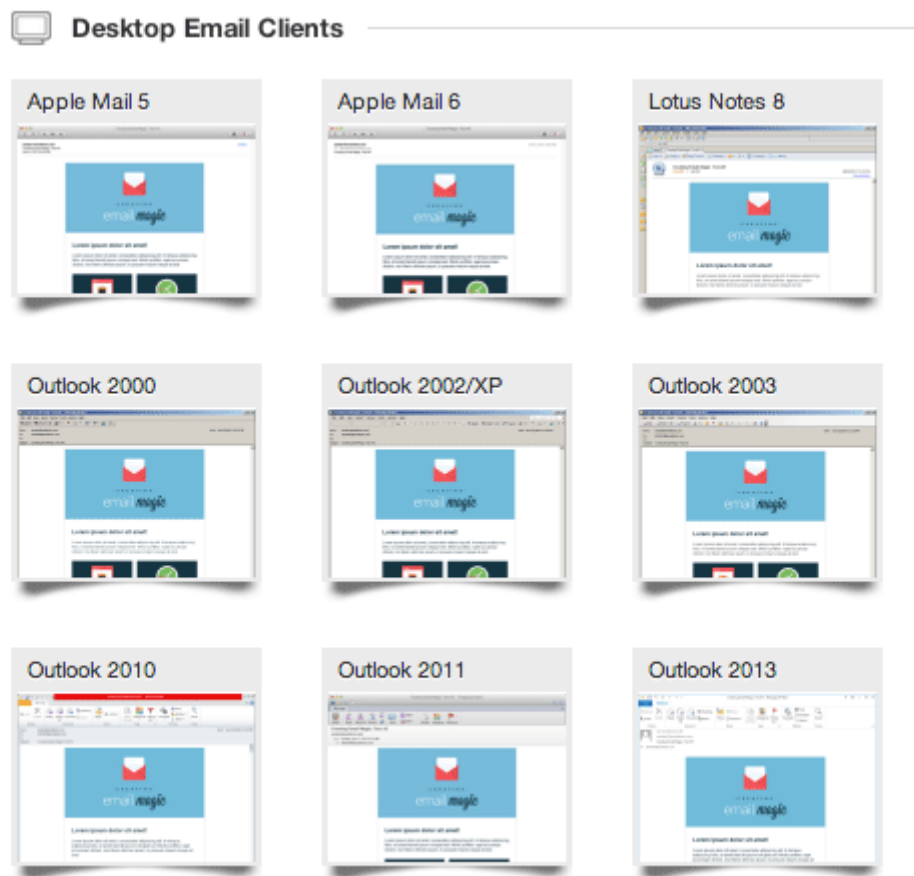


Lorem ipsum dolor sit amet, consectetur adipiscing elit. In tempus adipiscing felis, sit amet blandit ipsum volutpat sed. Morbi porttitor, eget accumsan dictum, nisi libero ultricies ipsum, in posuere mauris neque at erat.

© Someone, somewhere 2013  
[Unsubscribe](#) to this newsletter instantly



And that's it! You're all ready for a final test.



**That's a Wrap!**

Before we call it a day, if you have used any comments, get rid of them. Some email clients can choke on comments so it's wisest not to have any unnecessary code anywhere in your files.

Now is a good time to do a final test, and then, your HTML email is ready to send!